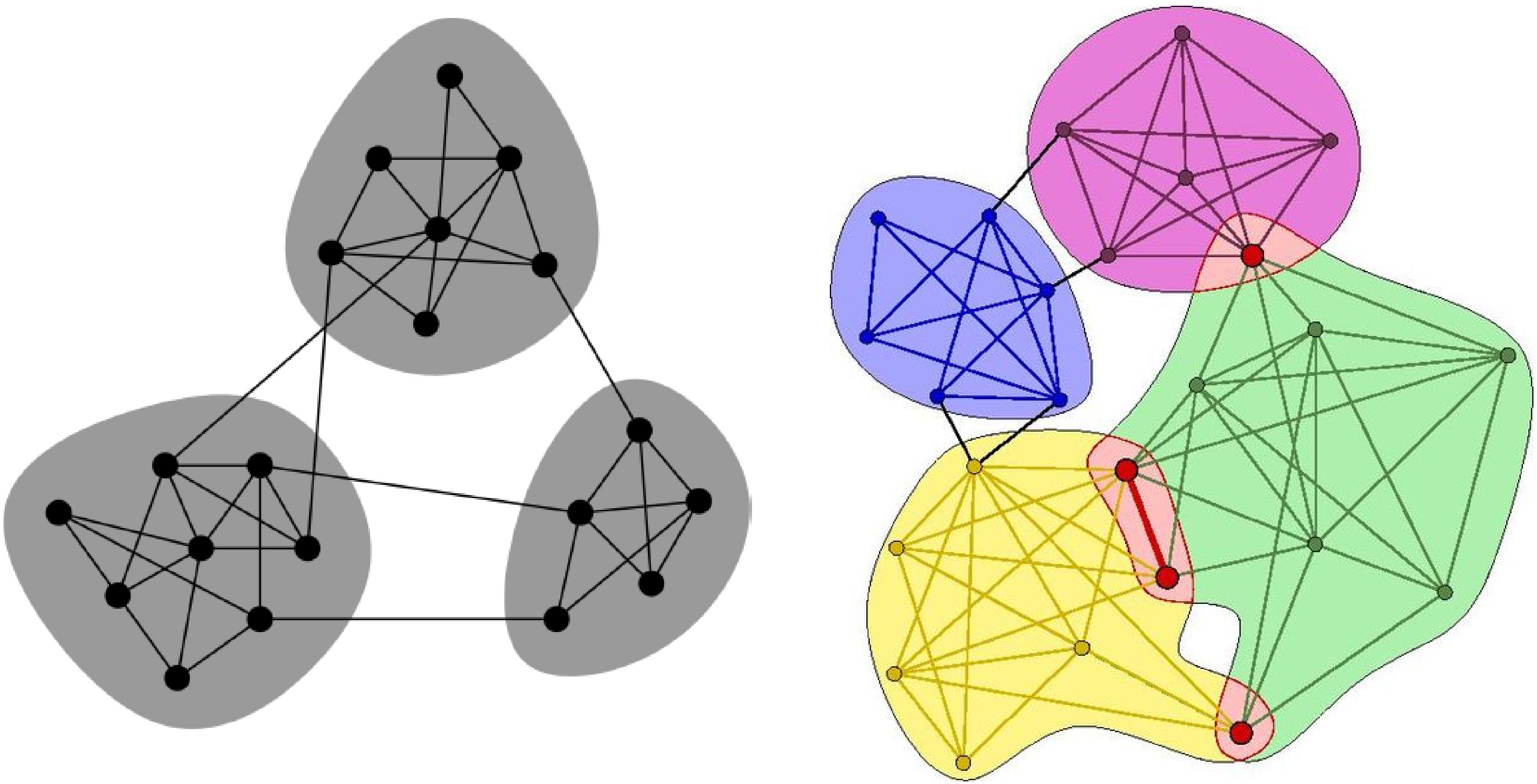
Community Detection: Overlapping Communities

CS224W: Social and Information Network Analysis
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

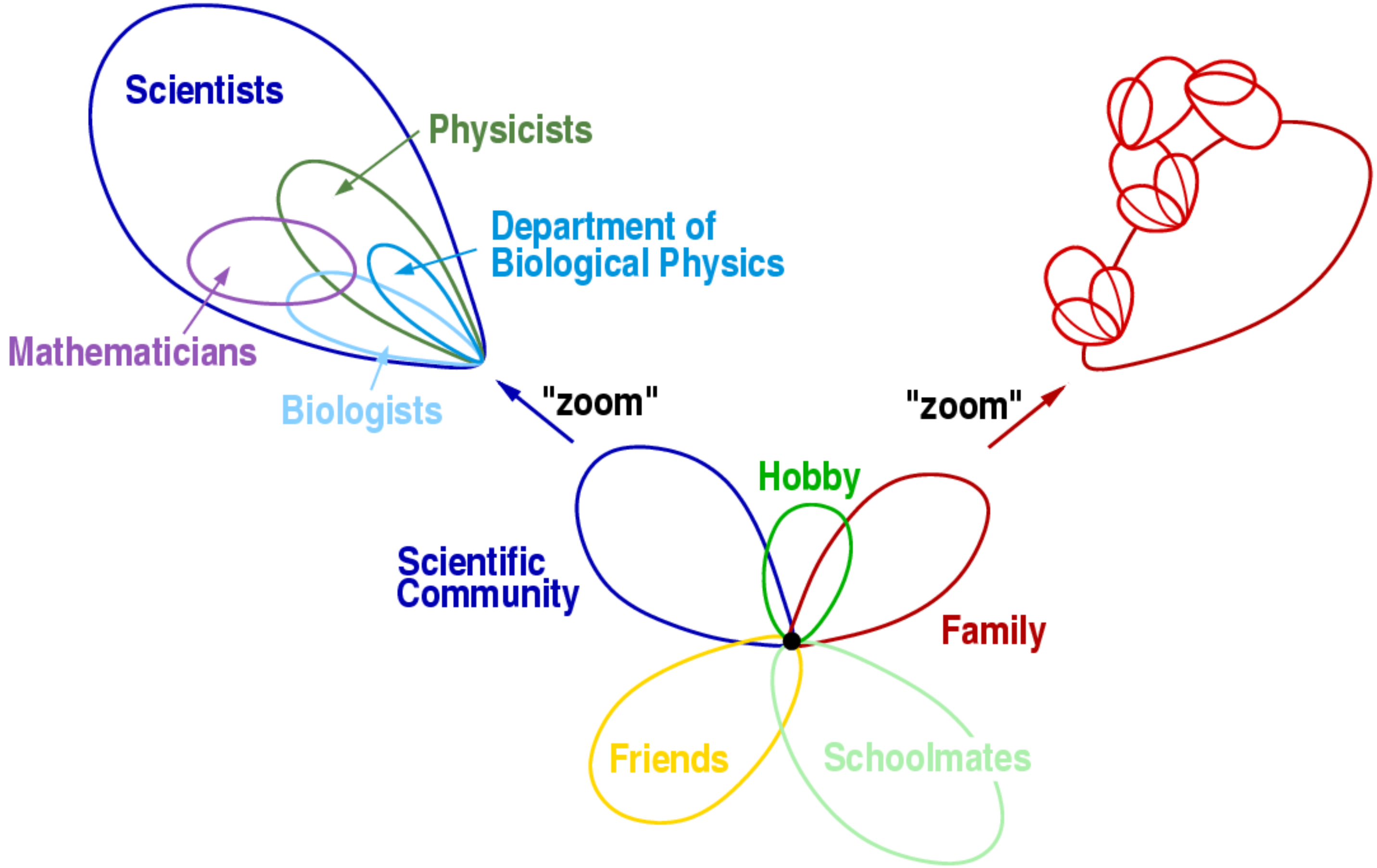
Overlapping Communities

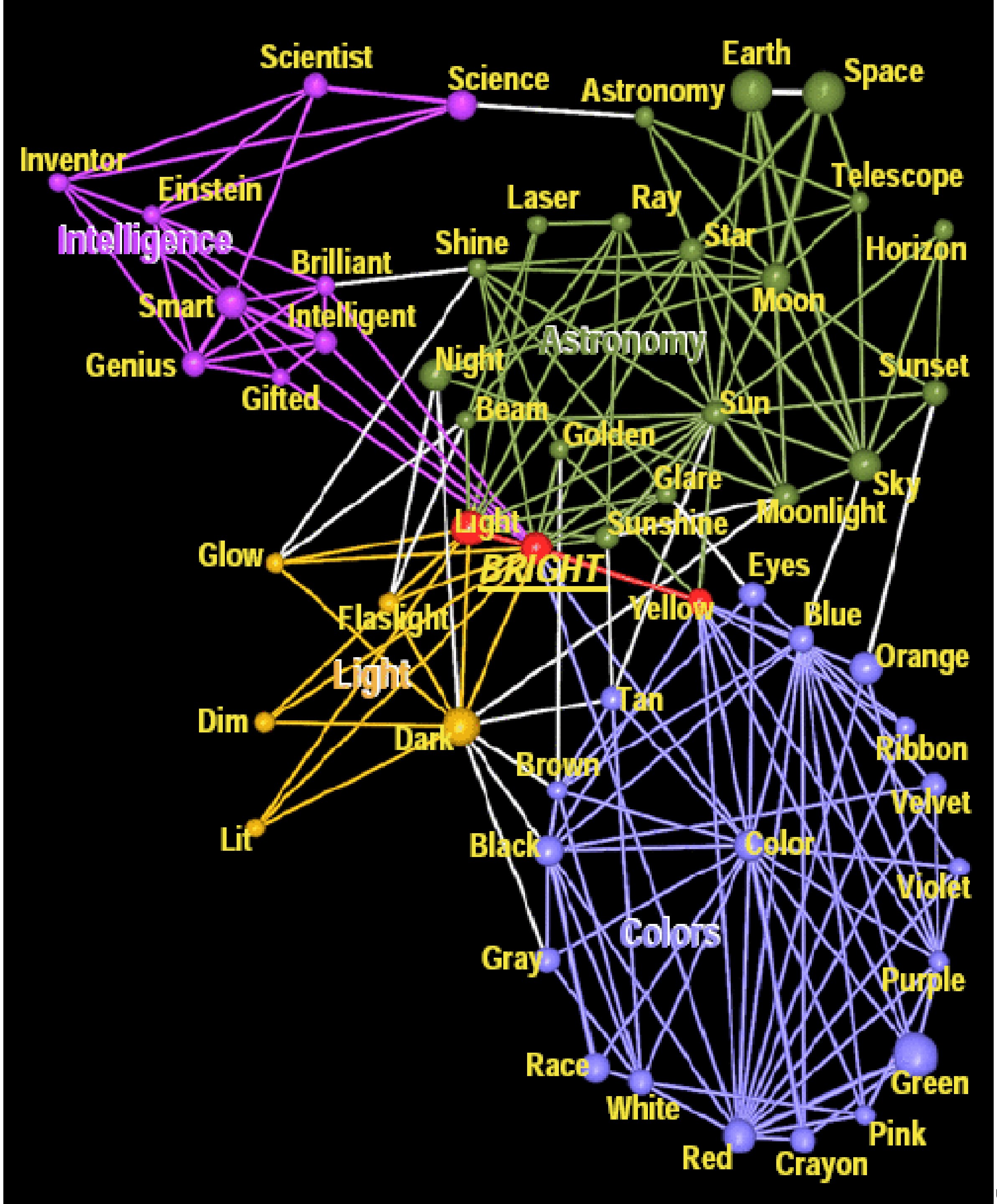
- **Non-overlapping vs. overlapping communities**



Overlaps of Social Circles

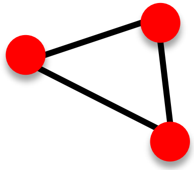
- A node belongs to many social circles

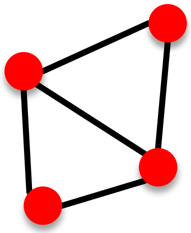


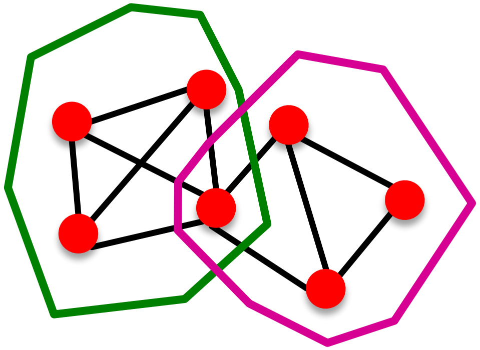

Scientist
Science
Astronomy
Earth
Space
Inventor
Einstein
Intelligence
Telescope
Laser
Ray
Shine
Star
Horizon
Brilliant
Moon
Smart
Intelligent
Astronomy
Genius
Night
Sunset
Gifted
Beam
Sun
Golden
Glare
Sky
Light
Sunshine
Moonlight
Glow
BRIGHT
Eyes
Yellow
Blue
Flaslight
Orange
Light
Tan
Dim
Dark
Ribbon
Brown
Velvet
Lit
Black
Color
Violet
Colors
Gray
Purple
Race
Green
White
Pink
Red
Crayon

Clique Percolation Method (CPM)

- **Two nodes belong to the same community if they can be connected through adjacent *k*-cliques:**
  - ***k*-clique:**
    - Fully connected graph on k nodes
  - **Adjacent *k*-cliques:**
    - overlap in k-1 nodes
- ***k*-clique community**
  - Set of nodes that can be reached through a sequence of adjacent k-cliques



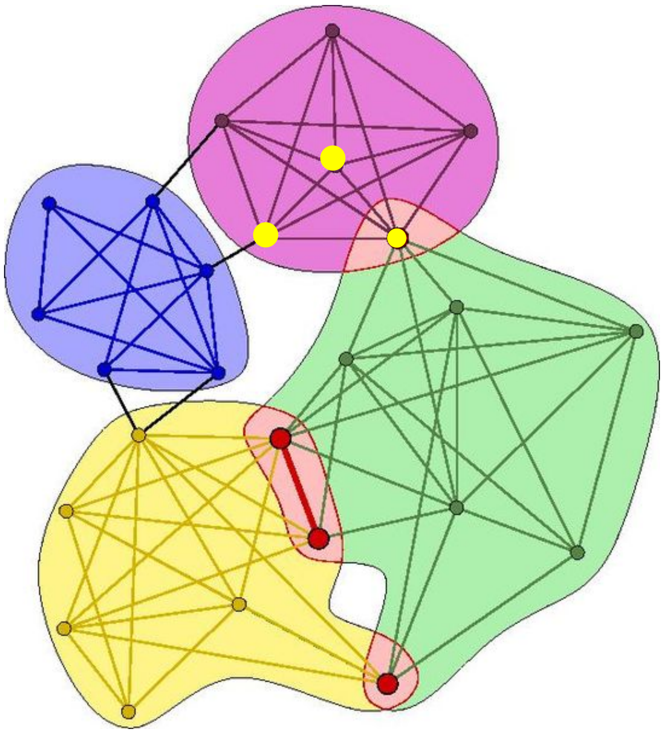




Clique Percolation Method (CPM)

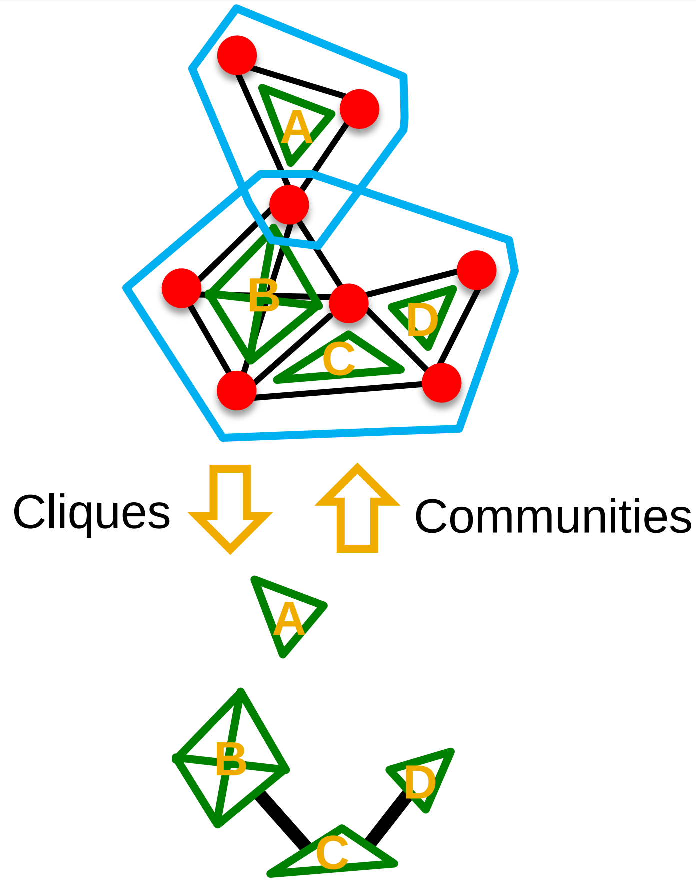
- **Two nodes belong to the same community if they can be connected through adjacent *k*-cliques:**

4-clique



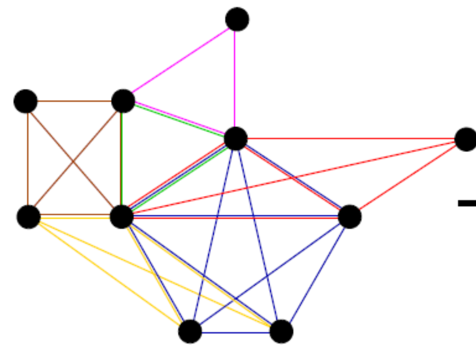
CPM: Steps

- **Clique Percolation Method:**
  - **Find maximal-cliques** (not k-cliques!)
  - **Clique overlap graph:**
    - Each clique is a node
    - Connect two cliques if they overlap in at least k-1 nodes
  - **Communities:**
    - Connected components of the clique overlap matrix
- **How to set k?**
  - Set k so that we get the "richest" (most widely distributed cluster sizes) community structure



CPM method: Example

- Start with graph
- Find maximal cliques
- Create clique overlap matrix
- Threshold the matrix at value *k-1*
  - If $a_{ij} < k\text{-}1$ *set 0*
- Communities are the connected components of the thresholded matrix



(1) Graph

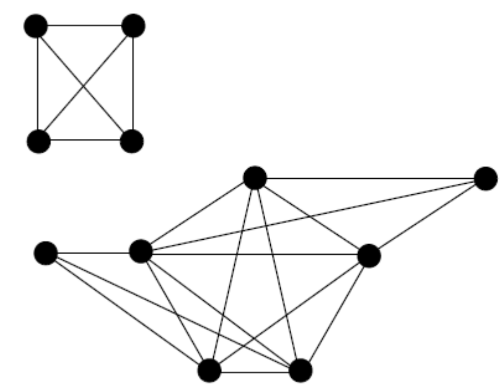
	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

(2) Clique overlap matrix

k=4

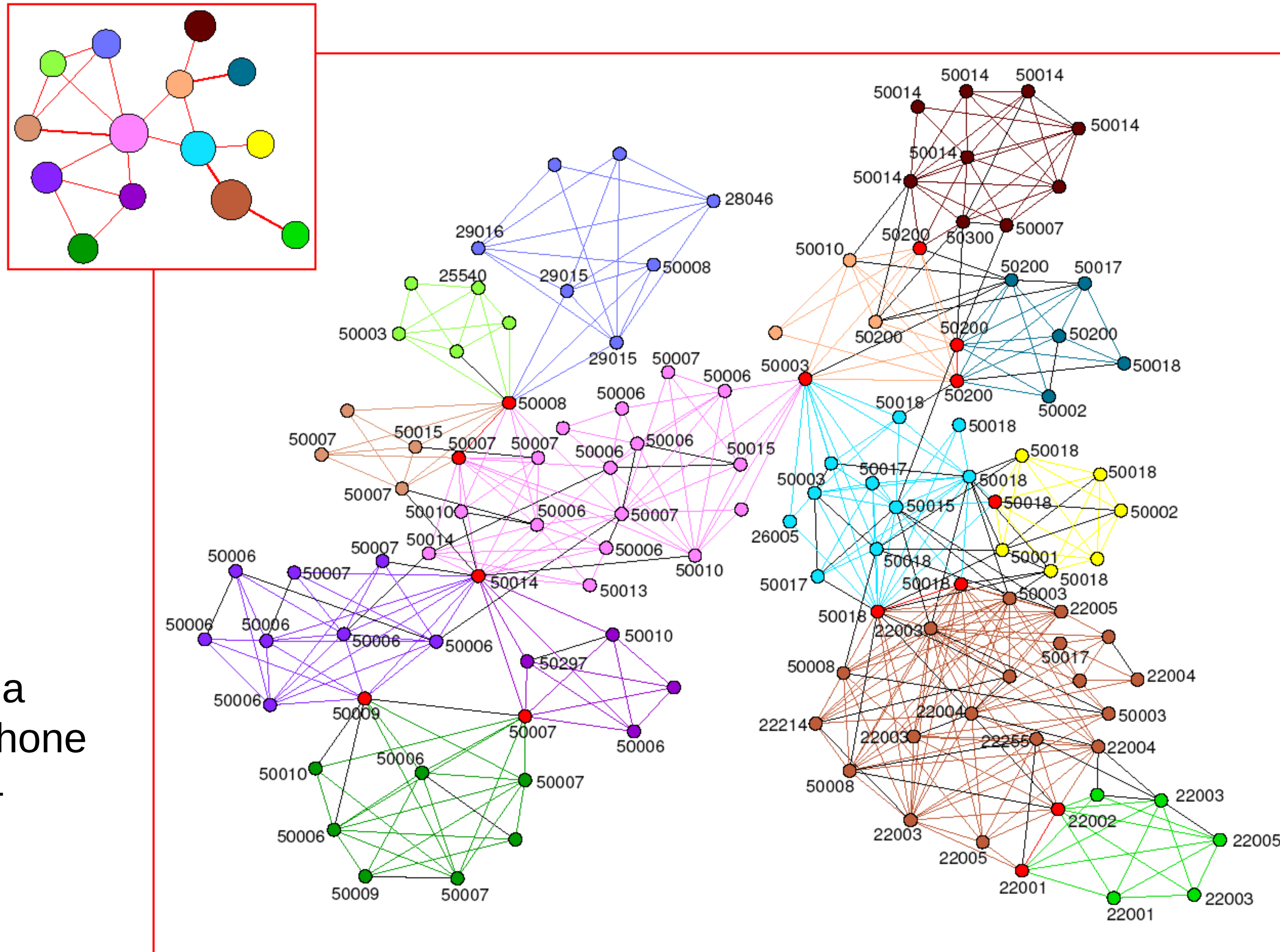
	1	1	0	0	1	0
	1	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	1	0
	0	0	0	0	0	1

(3) Thresholded matrix at 3



(4) Communities (connected components)

Example: Phone-Call Network



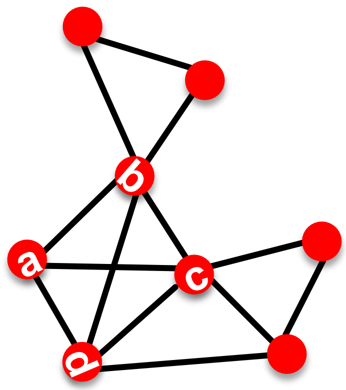
Communities in a “tiny” part of a phone call network of 4 million users [Palla et al., ‘07]

[Farkas et. al. 07]

Example: Website

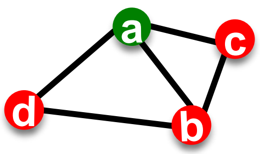
How to Find Maximal Cliques?

- No nice way, NP-hard combinatorial problem
- **Maximal clique:** clique that can't be extended
  - {a,b,c} is a clique but not maximal clique
  - {a,b,c,d} is maximal clique
- **Algorithm:** Sketch
  - Start with a seed node
  - Expand the clique around the seed
  - Once the clique cannot be further expanded we found the maximal clique
  - **Note:**
    - This will generate the same clique multiple times



How to Find Maximal Cliques?

- Start with a seed vertex "a"
- **Goal:** Find the maximal clique Q "a" belongs to
  - **Observation:**
    - If some "x" belongs to Q then it is a member of "a"
      - **Why?** If a,x $\in$ Q but not a–x, then Q is not a clique!
- **Recursive algorithm:**
  - Q … current clique
  - R … candidate vertices to expand the clique to
- **Example:** Start with "a" and expand around it



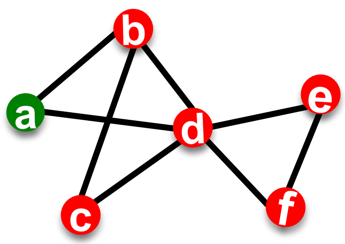
Q=	{a}	{a,b}	{a,b,c}	bktrack	{a,b,d}
R=	{$\underline{b}$,c,d}	{b,c,d} $\cap\Gamma$(b)={$\underline{c}$,d}	{d}$\cap\Gamma$(c)={}		{c}$\cap\Gamma$(d)={}

Steps of the recursive algorithm

Γ(u)…neighbor set of u

How to Find Maximal Cliques?

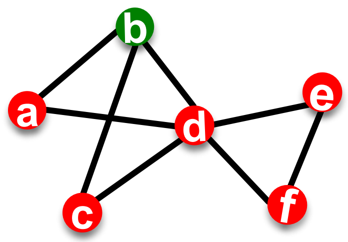
- Q … current clique
- R … candidate vertices
- **Expand(R,Q)**
  - **while** $R \neq \{\}$
    - p = vertex in R
    - $Q_p = Q \cup \{p\}$
    - $R_p = R \cap \Gamma(p)$
    - **if** $R_p \neq \{\}$: Expand(R_p,Q_p)
      else: output Q_p
    - $R = R - \{p\}$



Start: Expand(V, {})
R={a,…f}, Q={}
p = {a}
Q_p = {a}
R_p = {b,d}
Expand(R_p, Q):
R = {b,d}, Q={a}
p = {b}
Q_p = {a,b}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={a,b}
p = {d}
Q_p = {a,b,d}
R_p = {} : **output {a,b,d}**
p = {d}
Q_p = {a,d}
R_p = {b}
Expand(R_p, Q):
R = {b}, Q={a,d}
p = {b}
Q_p = {a,d}
R_p = {} : **output {a,d,b}**

How to Find Maximal Cliques?

- Q … current clique
- R … candidate vertices
- **Expand(R,Q)**
  - **while** $R \neq \{\}$
    - p = vertex in R
    - $Q_p = Q \cup \{p\}$
    - $R_p = R \cap \Gamma(p)$
    - **if** $R_p \neq \{\}$: Expand(R_p, Q_p)
      else: output Q_p
    - $R = R - \{p\}$

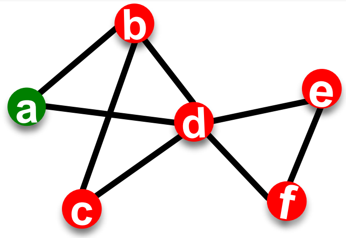


Start: Expand(V, {})

R={a,…f}, Q={}
p = {b}
Q_p = {b}
R_p = {a,c,d}
Expand(R_p, Q):
R = {a,c,d}, Q={b}
p = {a}
Q_p = {b,a}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={b,a}
p = {d}
Q_p = {b,a,d}
R_p = {} : **output {b,a,d}**
p = {c}
Q_p = {b,c}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={b,c}
p = {d}
Q_p = {b,c,d}
R_p = {} : **output {b,c,d}**

How to Find Maximal Cliques?

- **How to prevent maximal cliques to be generated multiple times?**
  - Only output cliques that are lexicographically minimum
    - {a,b,c} < {b,a,c}
  - **Even better:** Only expand to the nodes higher in the lexicographical order



Start: Expand(V, {})

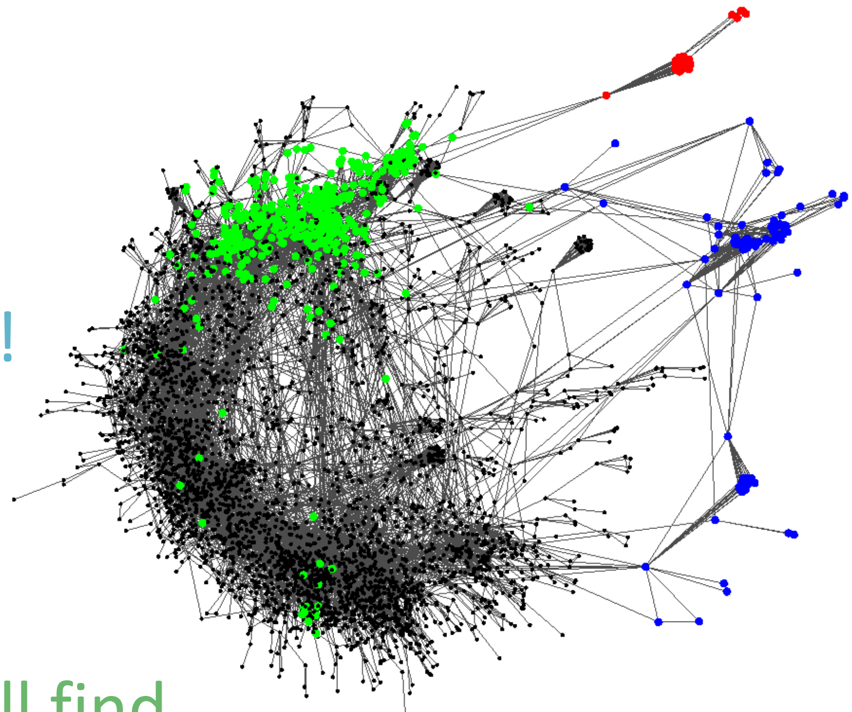
R={a,…f}, Q={}
p = {a}
Q_p = {a}
R_p = {b,d}
Expand(R_p, Q):
R = {b,d}, Q={a}
p = {b}
Q_p = {a,b}
R_p = {d}
Expand(R_p, Q):
R = {d}, Q={a,b}
p = {d}
Q_p = {a,b,d}
R_p = {} : **output {a,b,d}**
p = {d}
Q_p = {a,d}
R_p = {b}

Don't expand b < d

How to Model Networks with Communities?

Reflections: Finding Communities

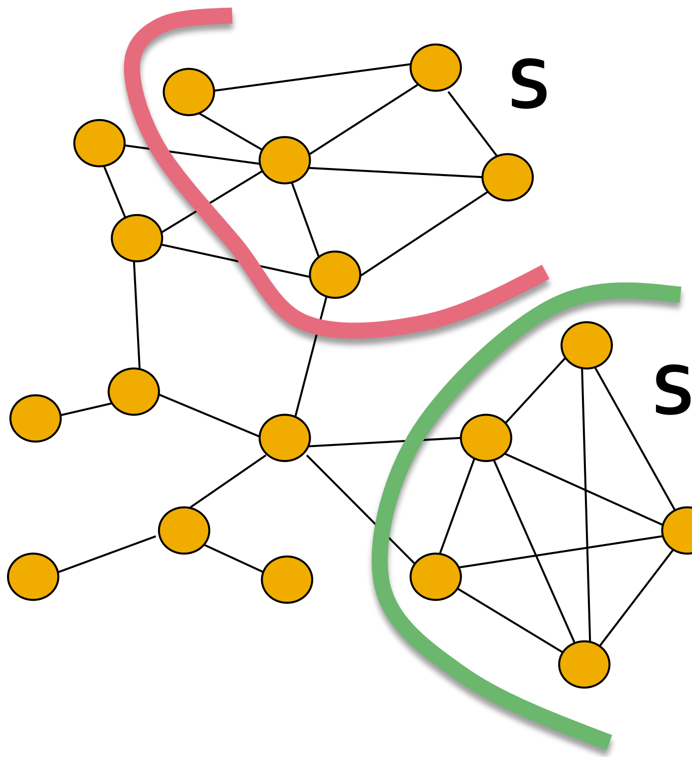
- **Let's rethink what we are doing…**
  - Given a network
  - Want to find communities!
- **Need to:**
  - Formalize the notion of a community
  - Need an algorithm that will find sets of nodes that are "good" communities
- **More generally:**
  - **How to think about clusters in large networks?**



Clustering Objective Functions

What is a good cluster?

- Many edges internally
- Few pointing outside



Formally, **conductance:**

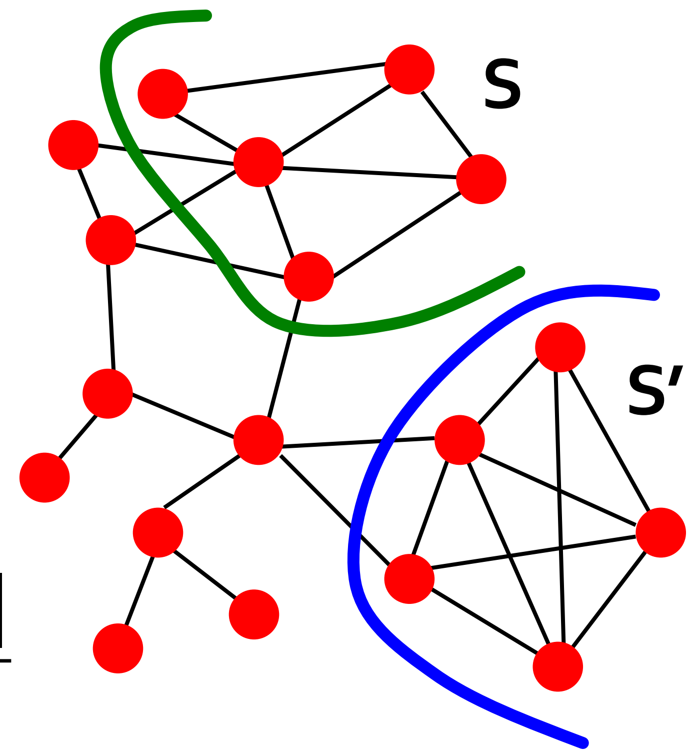
$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{A(S), A(\overline{S})\}}$$

Where: A(S)….volume $A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$

Small Φ(S) corresponds to good clusters

Community Score

- **How community like is a set of nodes?**
- A good cluster *S* has
  - Many edges internally
  - Few edges pointing outside
- Simplest objective function:
  Conductance



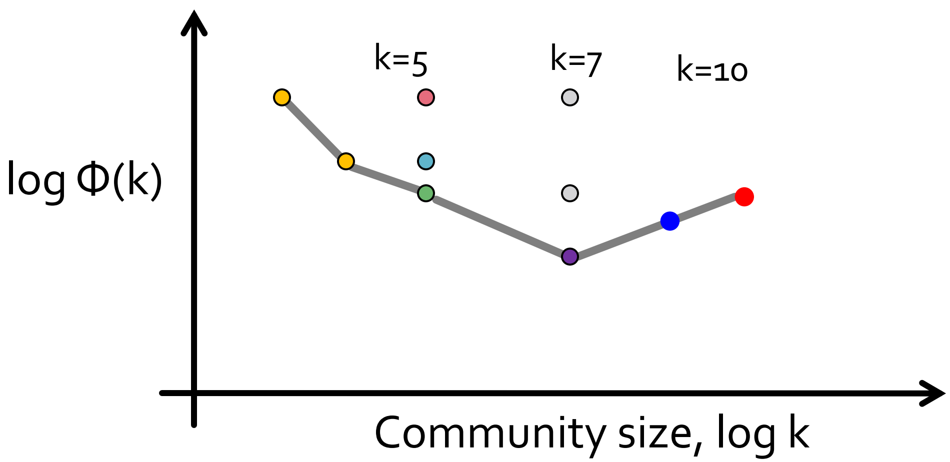
$$\phi(S) = \frac{|\{(i,j) \in E; i \in S, j \notin S\}|}{\sum_{s \in S} d_s}$$

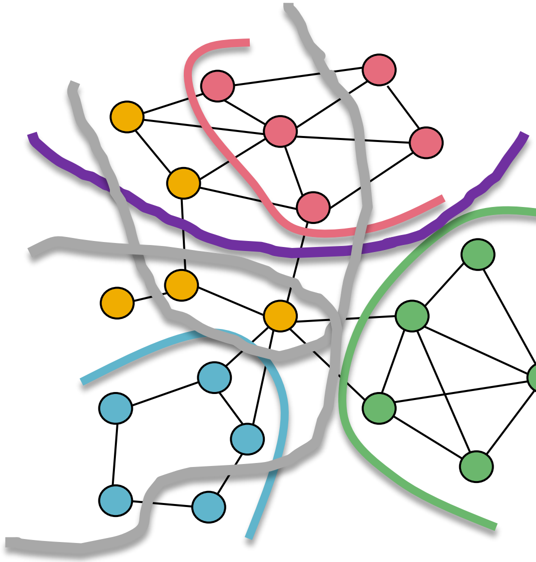
Small conductance corresponds to good clusters

Network Community Profile Plot

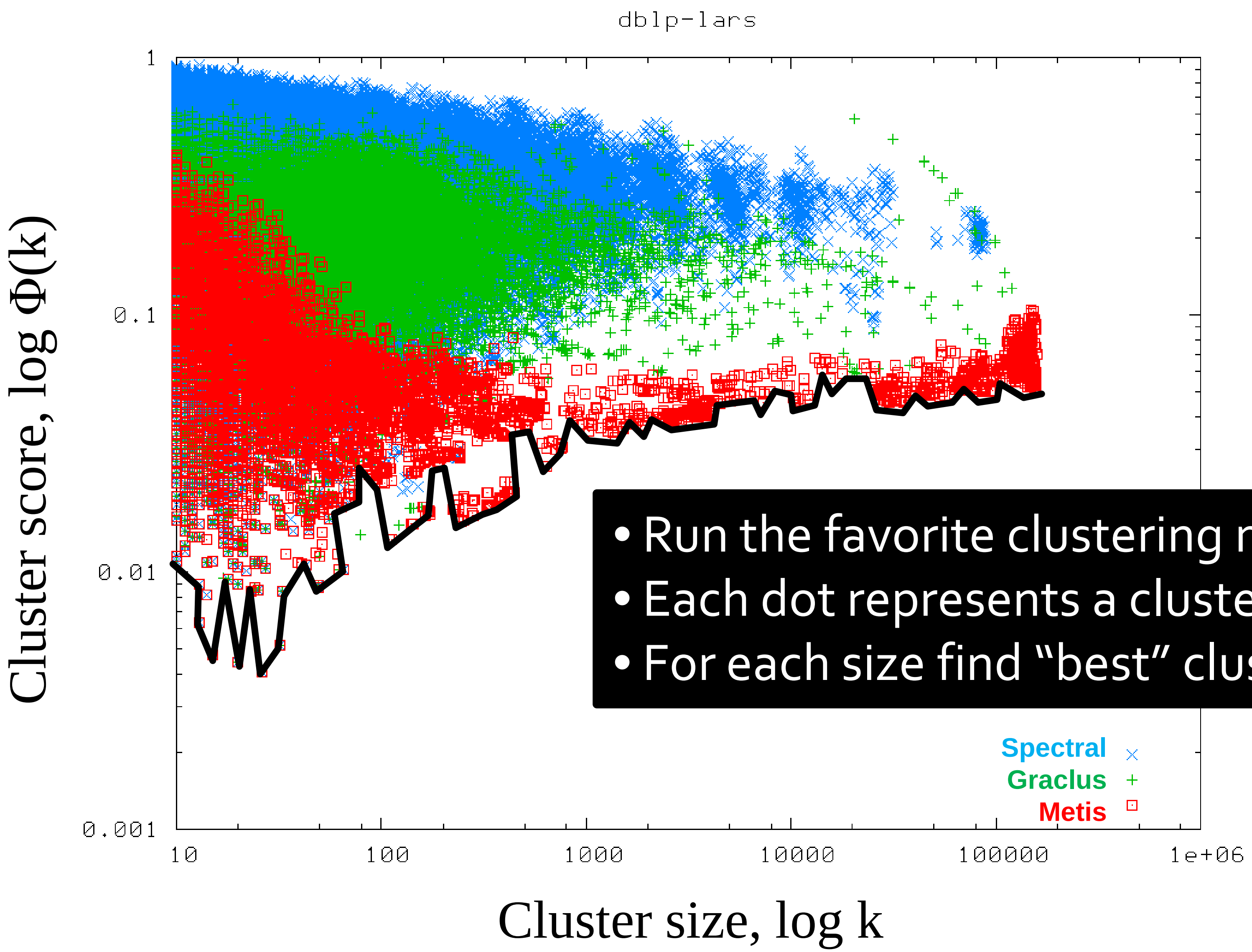
- **Define:**
  Network community profile (**NCP**) plot
  Plot the score of **best** community of size *k*

$$\Phi(k) = \min_{S \subset V, |S|=k} \phi(S)$$





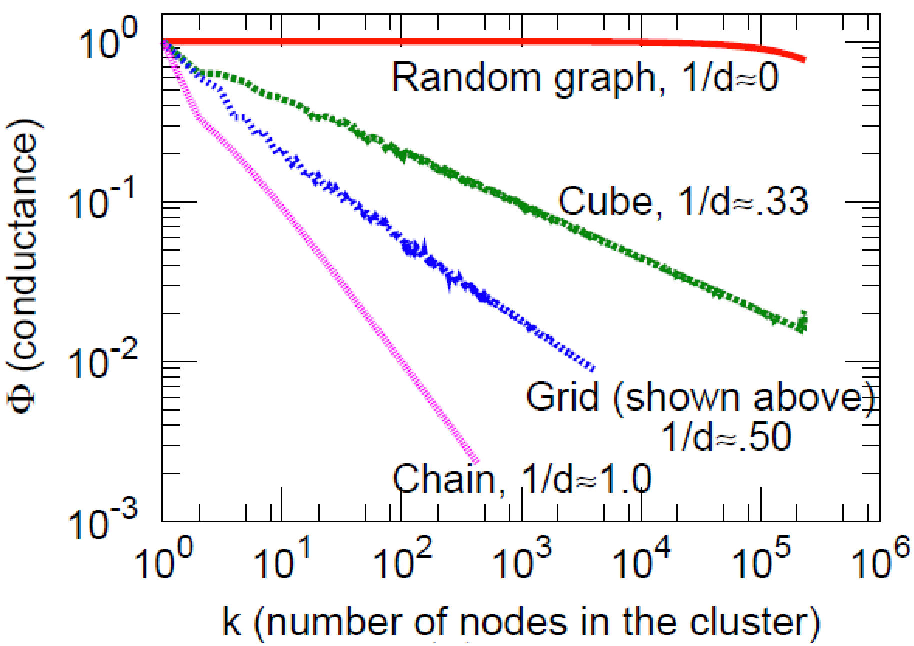
How to (Really) Compute NCP?



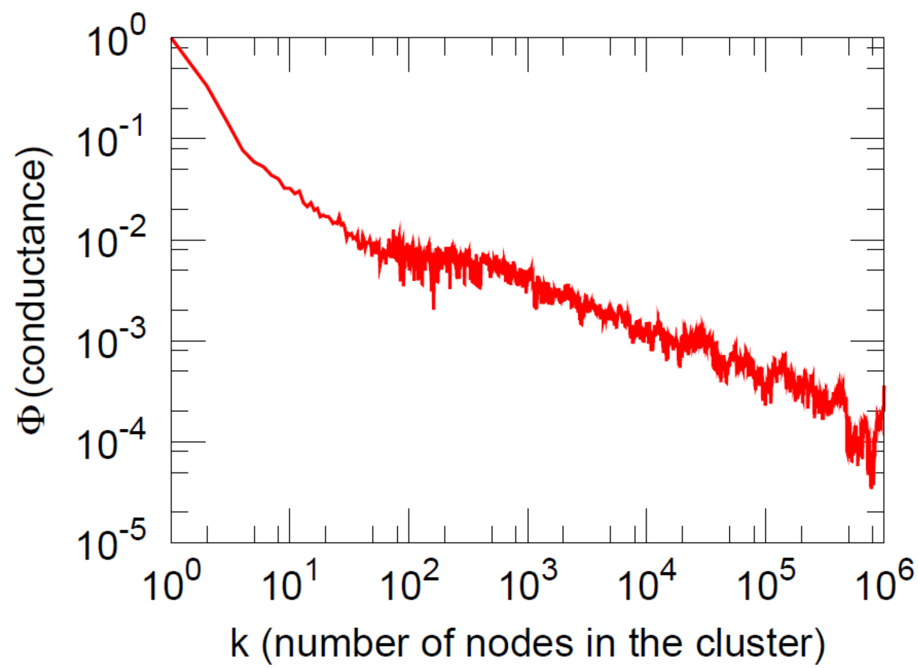
- Run the favorite clustering method
- Each dot represents a cluster
- For each size find “best” cluster

NCP Plot: Meshes

- Meshes, grids, dense random graphs:



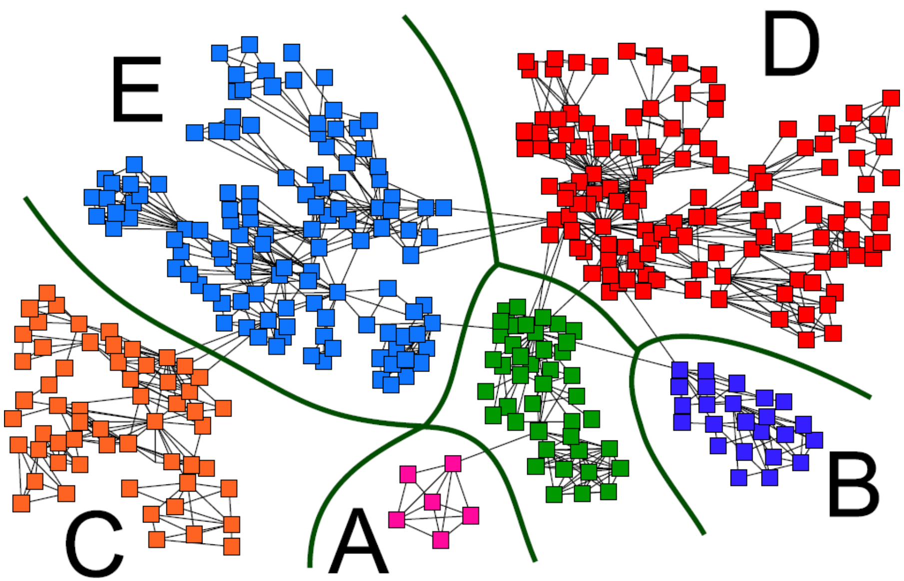
d-dimensional meshes

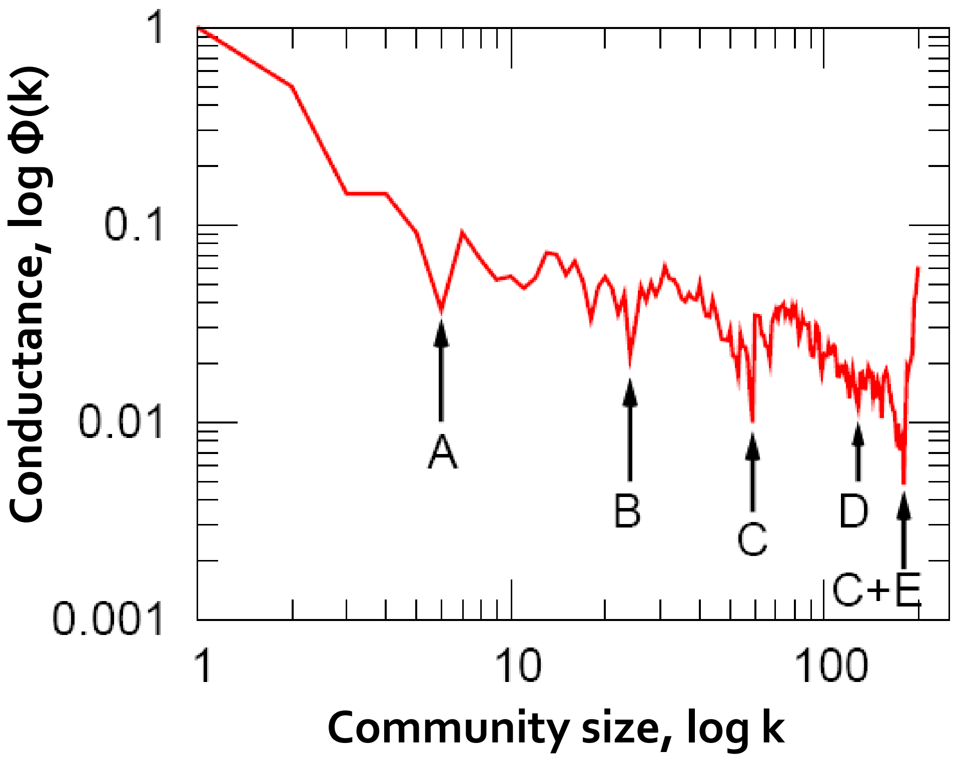


California road network

NCP plot: Network Science

- **Collaborations between scientists in networks**
  [Newman, 2005]

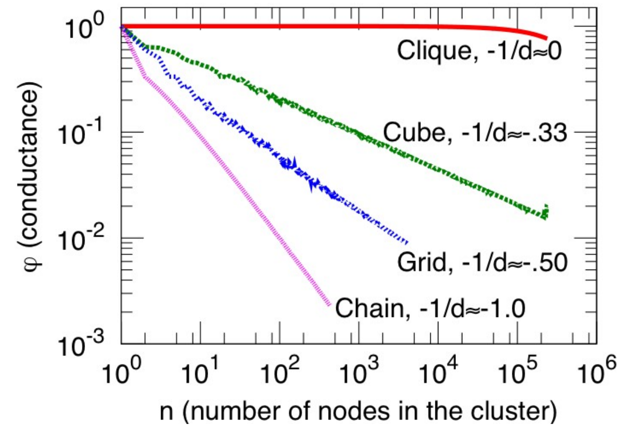




Natural Hypothesis

Natural hypothesis about NCP:

- NCP of real networks slopes downward
- Slope of the NCP corresponds to the "dimensionality" of the network

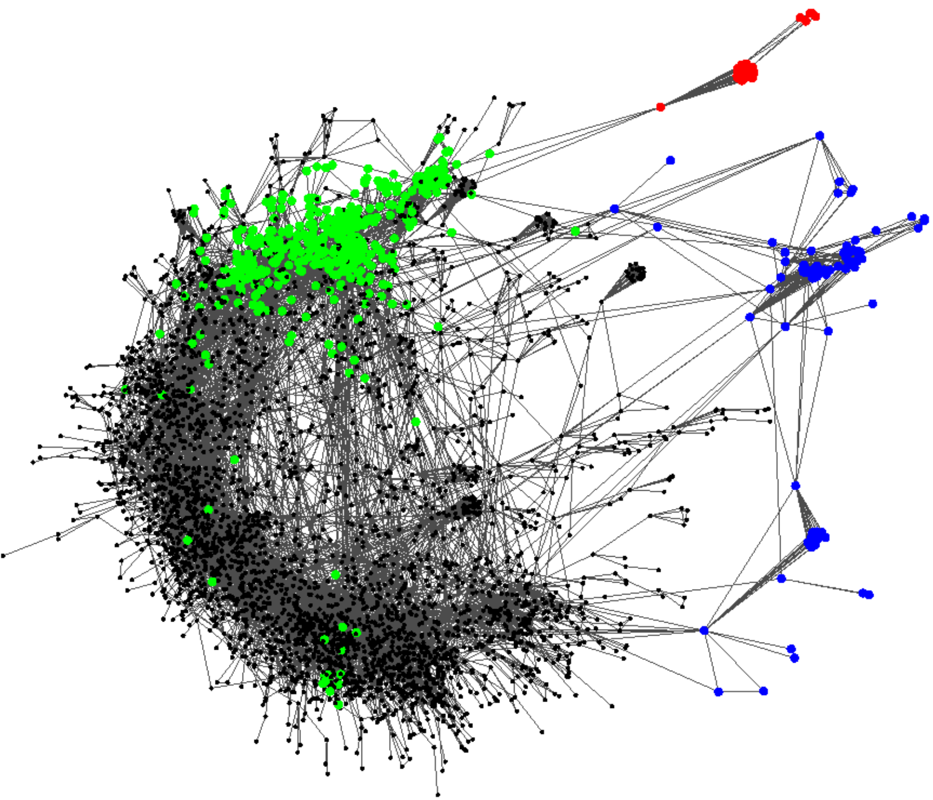


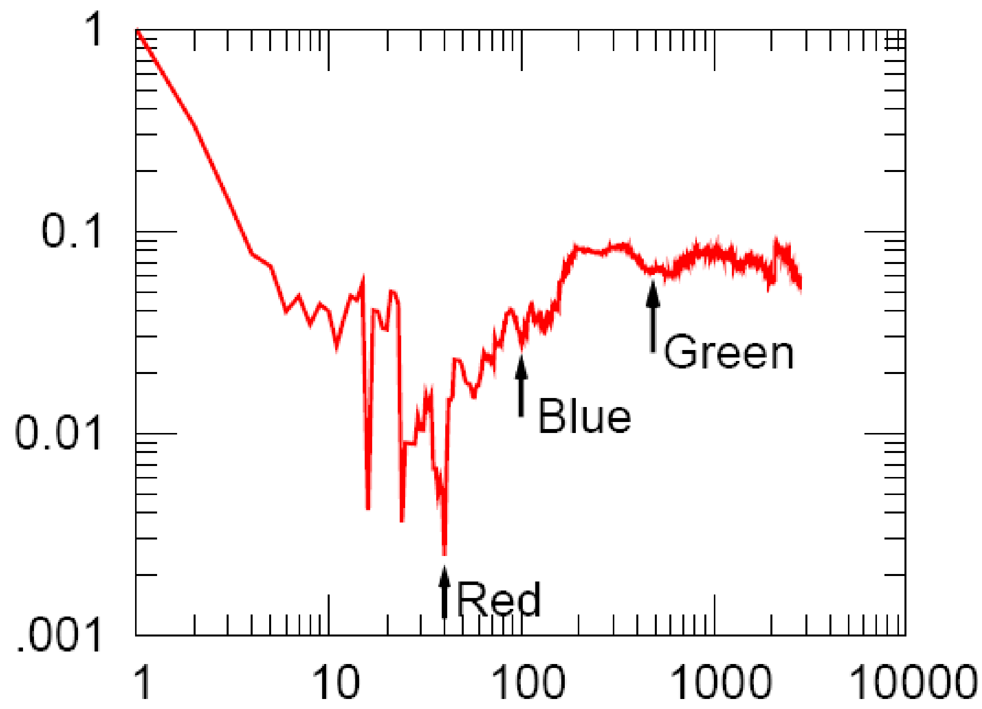
What about large networks?

• Social nets	Nodes	Edges	Description
LiveJournal	4,843,953	42,845,684	Blog friendships [5]
Epinions	75,877	405,739	Trust network [28]
CA-DBLP	317,080	1,049,866	Co-authorship [5]
• Information (citation) networks			
Cit-hep-th	27,400	352,021	Arxiv hep-th [14]
AmazonProd	524,371	1,491,793	Amazon products [8]
• Web graphs			
Web-google	855,802	4,291,352	Google web graph
Web-wt10g	1,458,316	6,225,033	TREC WT10G
• Bipartite affiliation (authors-to-papers) networks			
AtP-DBLP	615,678	944,456	DBLP [21]
AtM-Imdb	2,076,978	5,847,693	Actors-to-movies
• Internet networks			
AsSkitter	1,719,037	12,814,089	Autonom. sys.
Gnutella	62,561	147,878	P2P network [29]

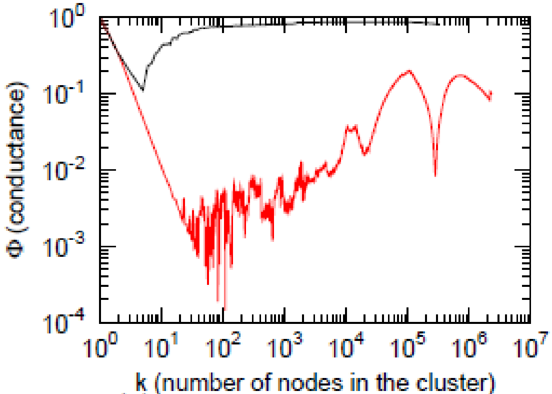
Large Networks: Very Different

Typical example: General Relativity collaborations (n=4,158, m=13,422)

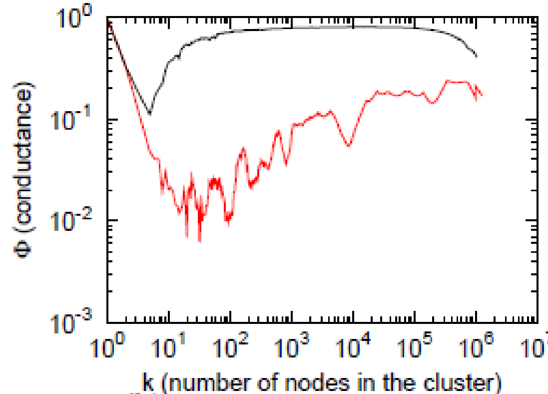




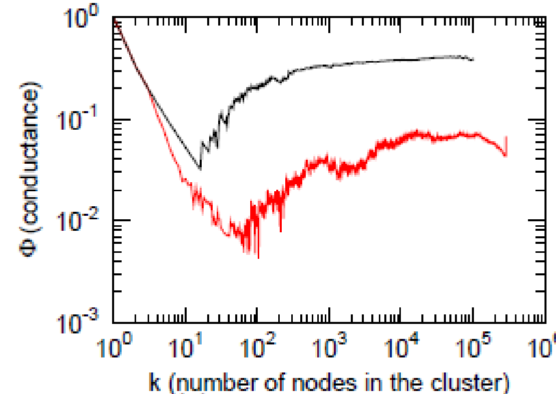
More NCP Plots of Networks



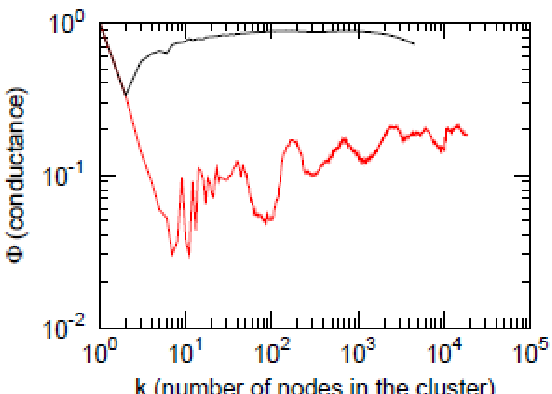
(a) LiveJournal01



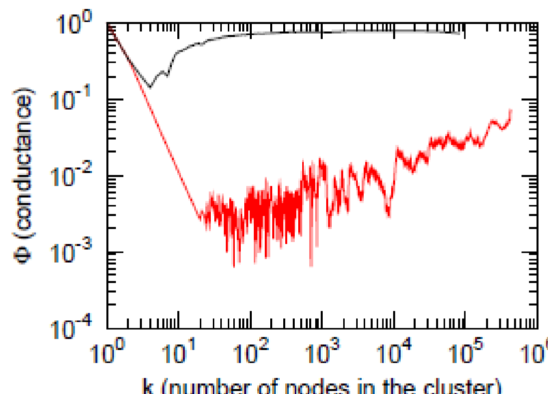
(b) Messenger-DE



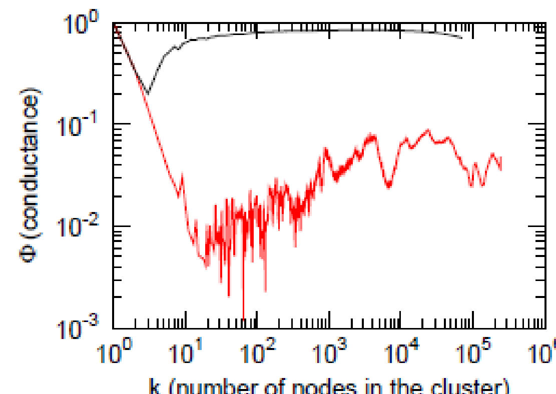
(c) AtP-DBLP



(d) Cit-hep-th

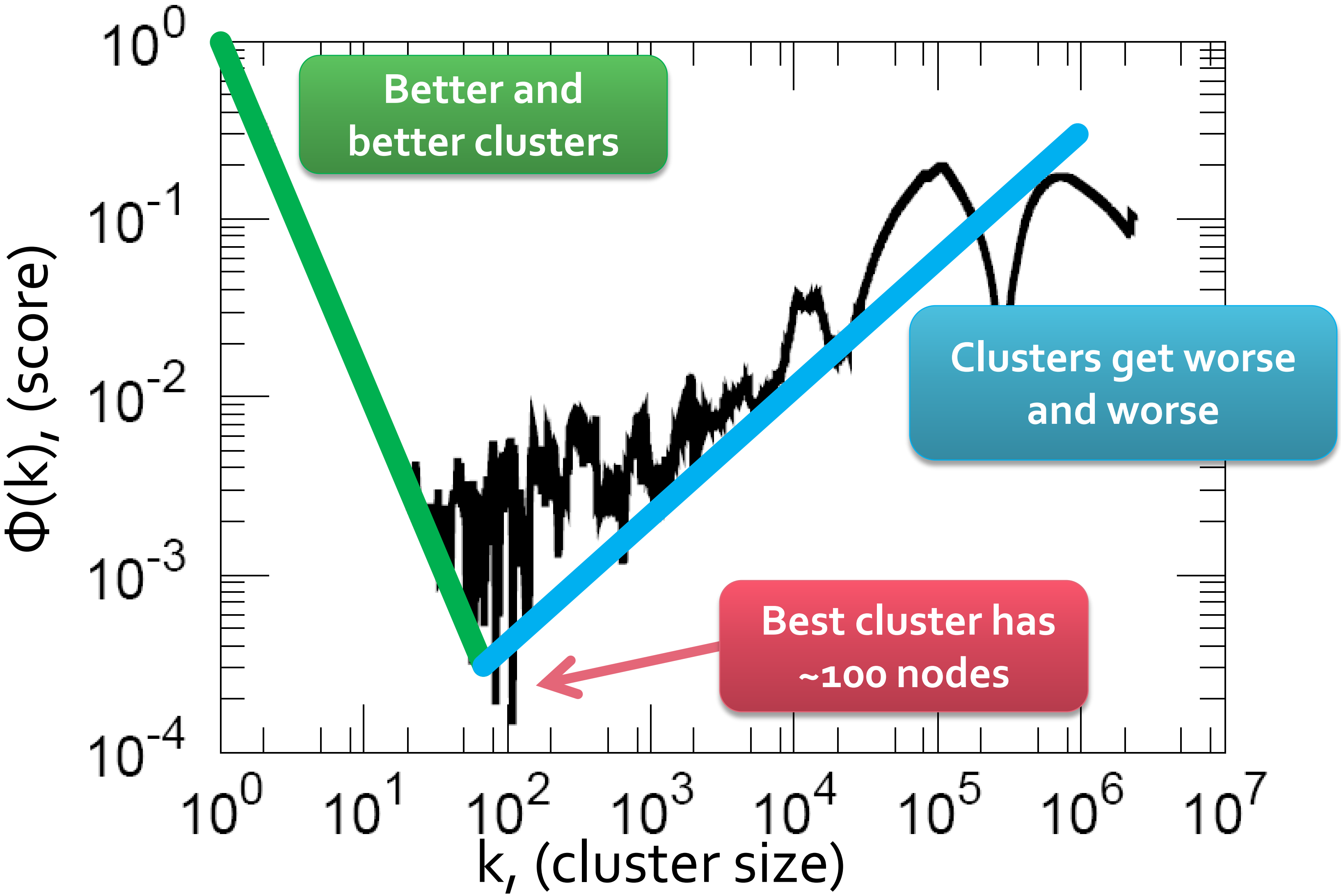


(e) Web-Google



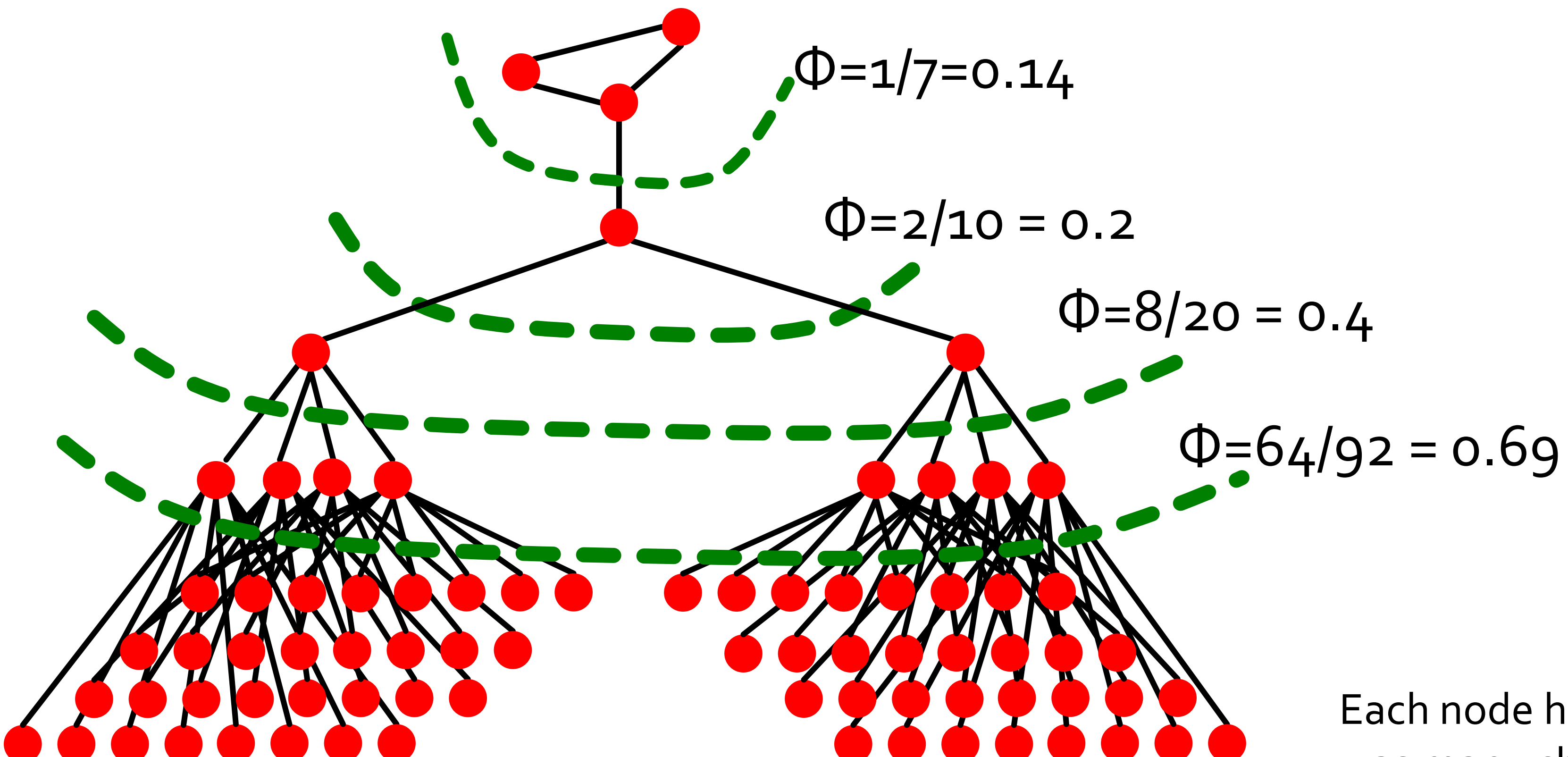
(f) AmazonAll

NCP: LiveJournal (n=5m, m=42m)



Explanation: The Upward Part

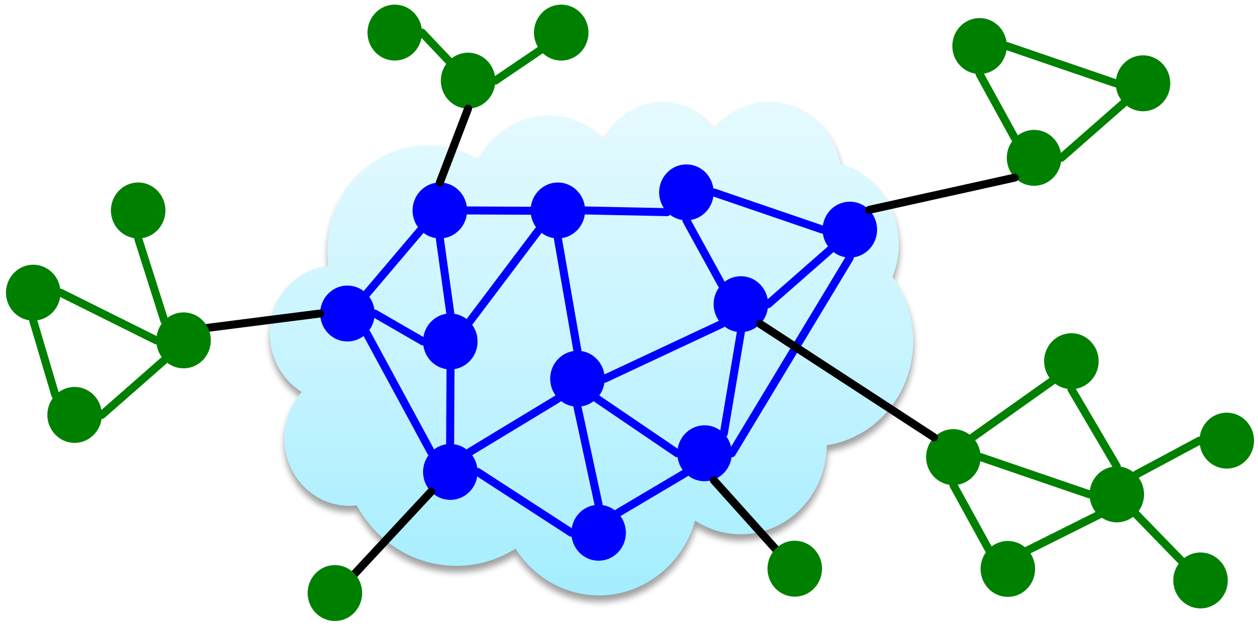
- As clusters grow the number of edges inside grows **slower** that the number crossing

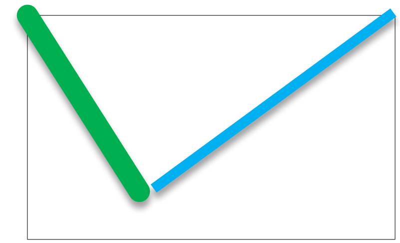


Each node has twice as many children

Explanation: Downward Part

- Empirically we note that best clusters are barely connected to the network

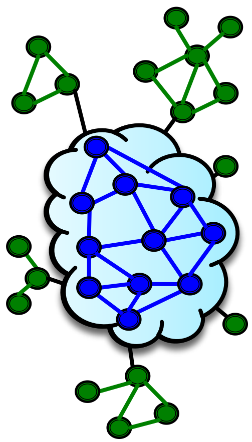


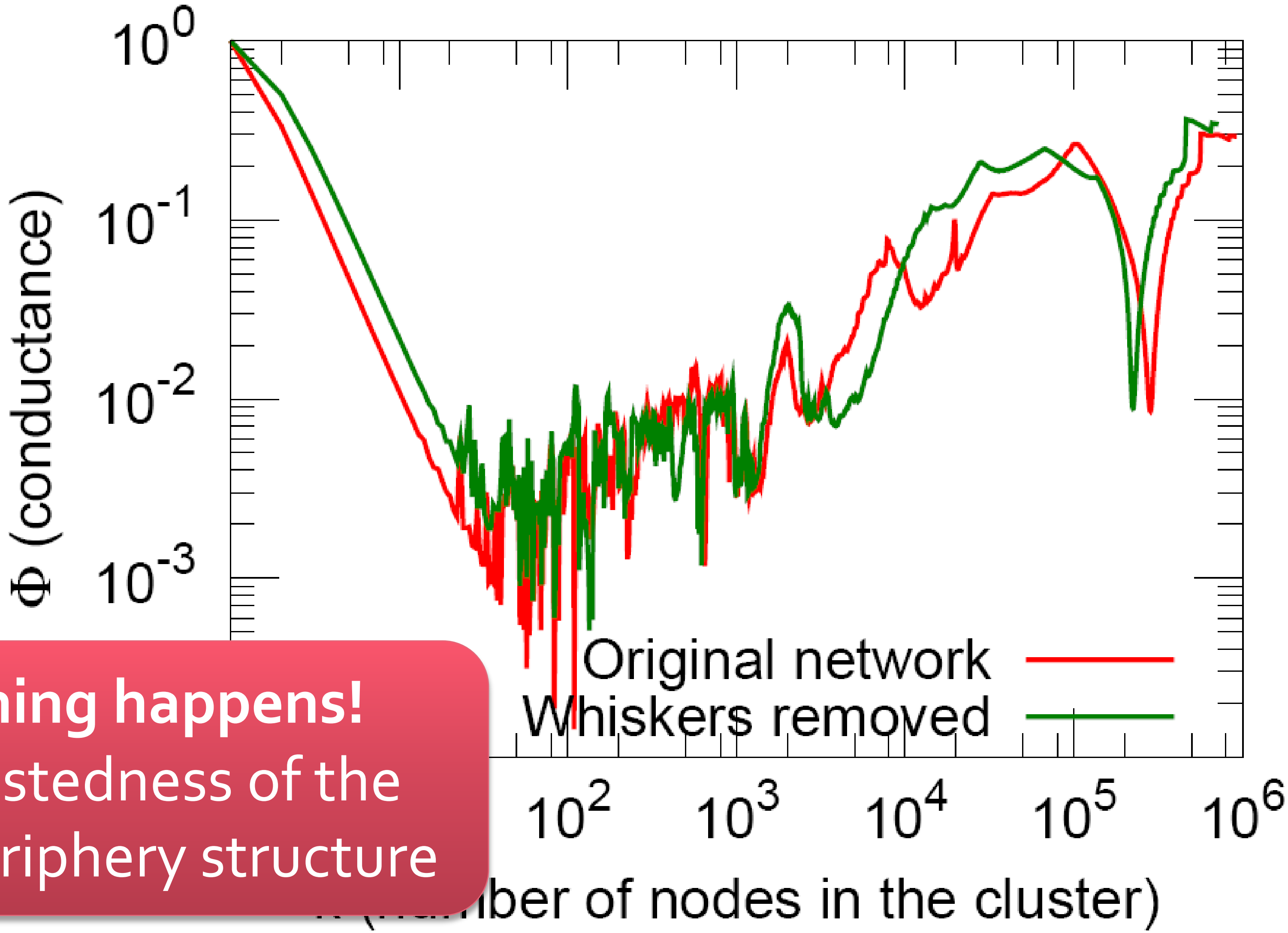


NCP plot

⇒ Core-periphery structure

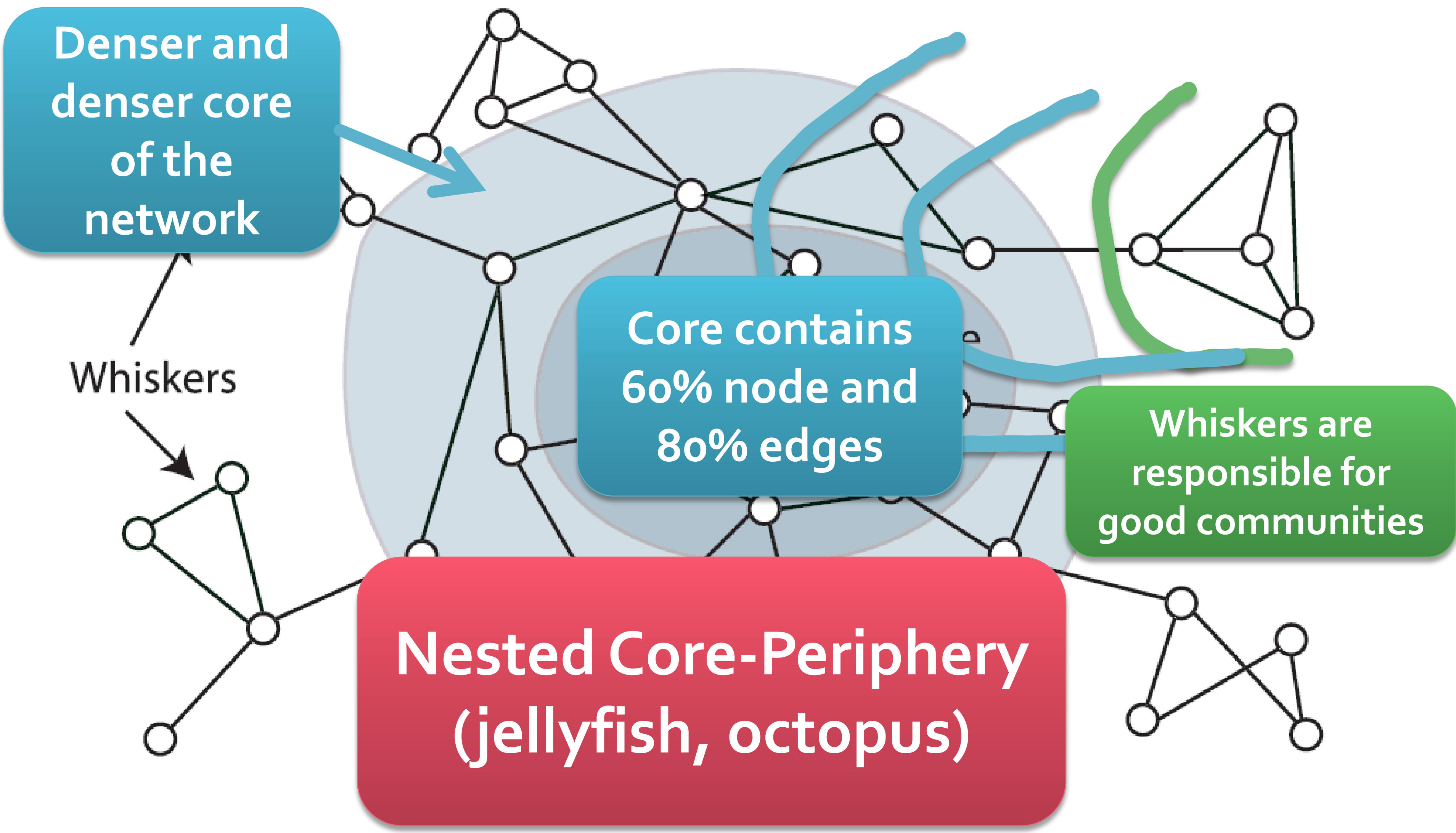
What If We Remove Good Clusters?





Nothing happens!
⇒ Nestedness of the core-periphery structure

Suggested Network Structure



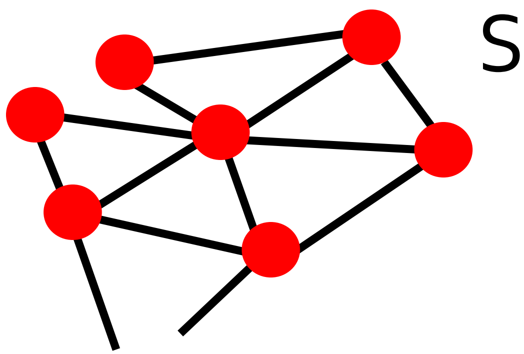
Communities: Issues and Questions

Communities: Issues and Questions

- **Some issues with community detection:**
  - Many different formalizations of clustering objective functions
  - Objectives are NP-hard to optimize exactly
  - Methods can find clusters that are systematically "biased"
- **Questions:**
  - **How well do algorithms optimize objectives?**
  - **What clusters do different methods find?**

Many Different Objective Functions

- **Single-criterion:**
  - Modularity: $m\text{-}E(m)$
  - Edges cut: c
- **Multi-criterion:**
  - Conductance: $c/(2m+c)$
  - Expansion: c/n
  - Density: $1\text{-}m/n^2$
  - CutRatio: $c/n(N\text{-}n)$
  - Normalized Cut: $c/(2m+c) + c/2(M\text{-}m)+c$
  - Flake-ODF: *frac. of nodes with more than ½ edges pointing outside S*



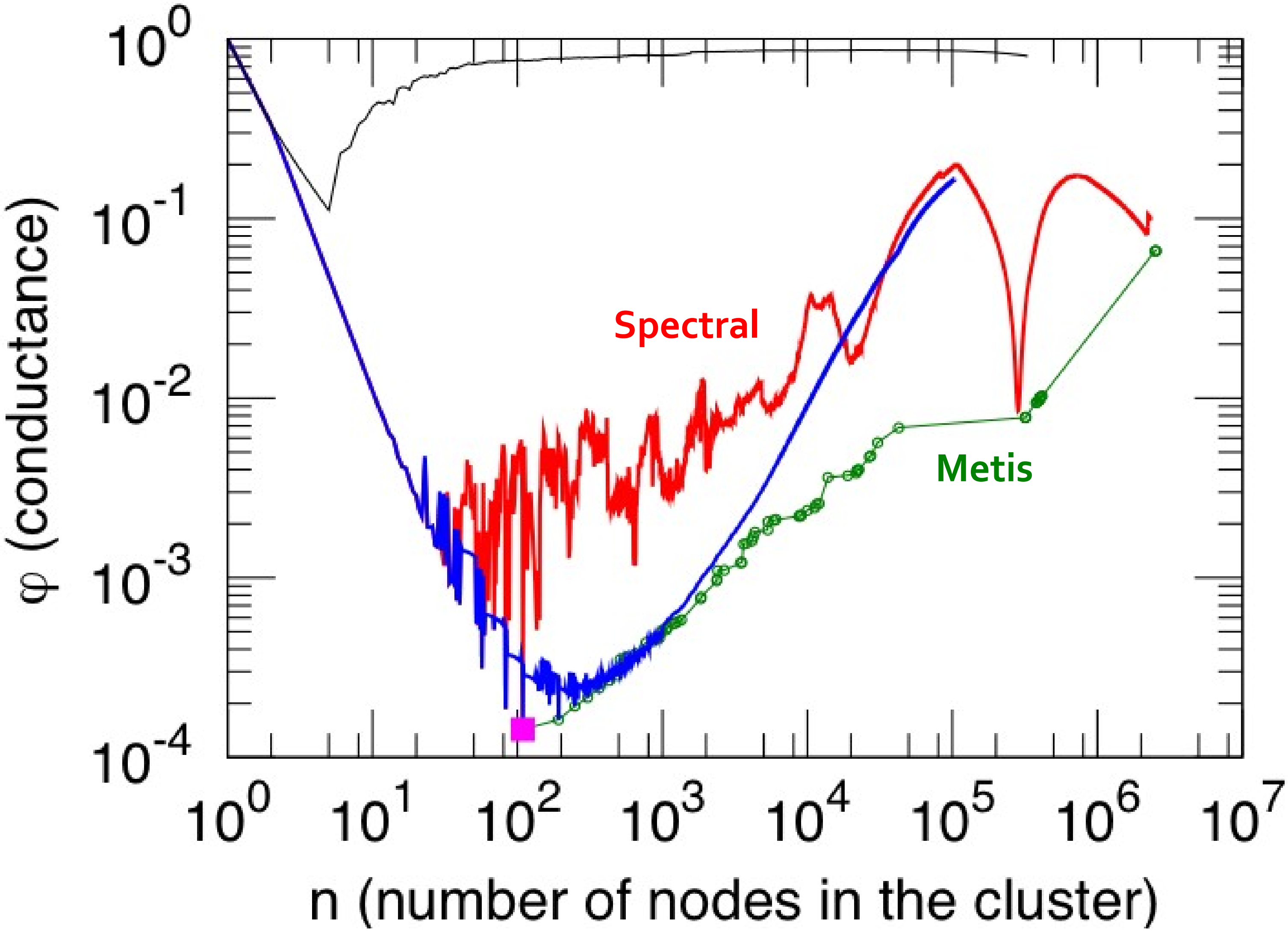
n: nodes in S
m: edges in S
c: edges pointing outside S

Many Classes of Algorithms

Many algorithms to that implicitly or explicitly optimize objectives and extract communities:

- **Heuristics:**
  - Girvan-Newman, Modularity optimization: popular heuristics
  - Metis: multi-resolution heuristic [Karypis-Kumar '98]
- **Theoretical approximation algorithms:**
  - Spectral partitioning

NCP: Live Journal



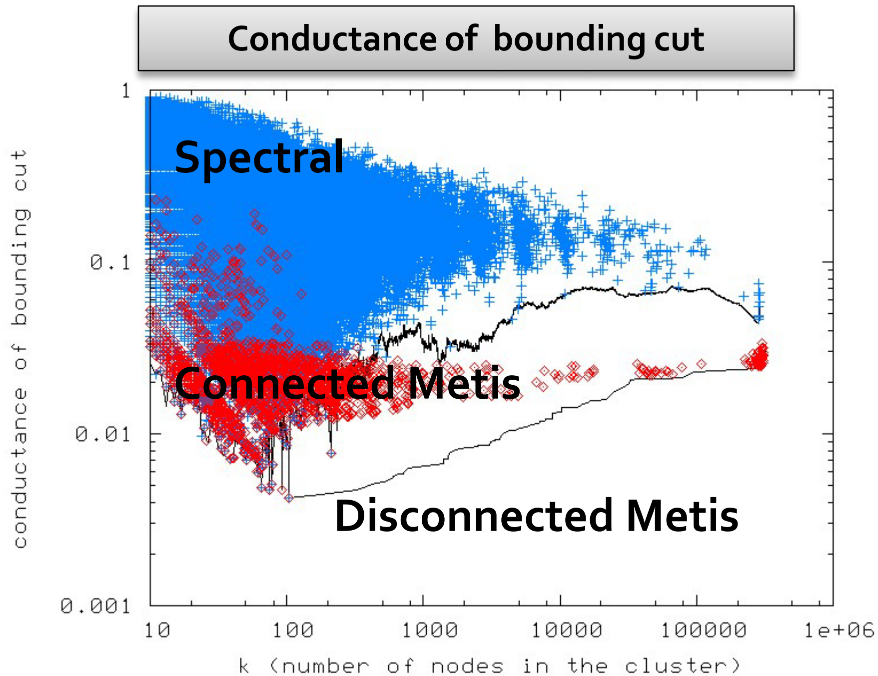
[WWW '09]

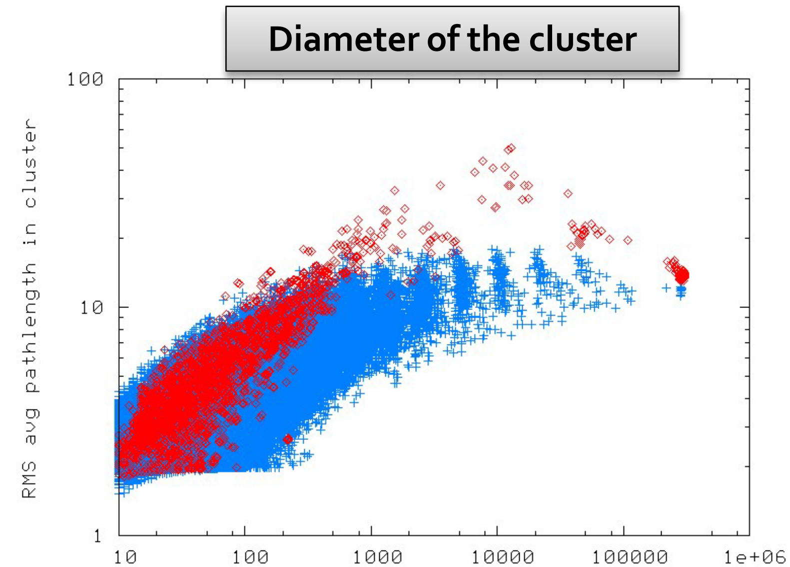
Properties of Clusters (1)

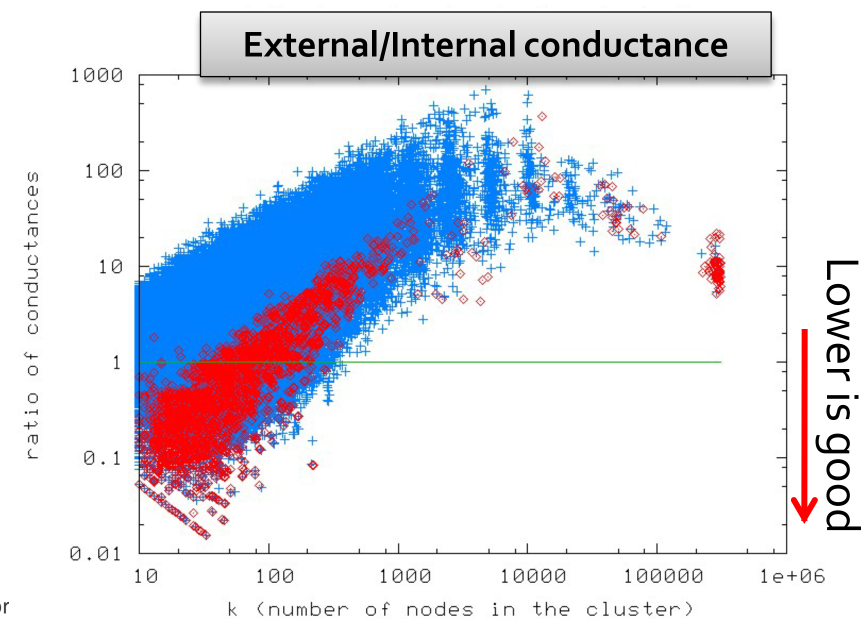
500 node communities from Spectral:

500 node communities from Metis:

Properties of Clusters (2)

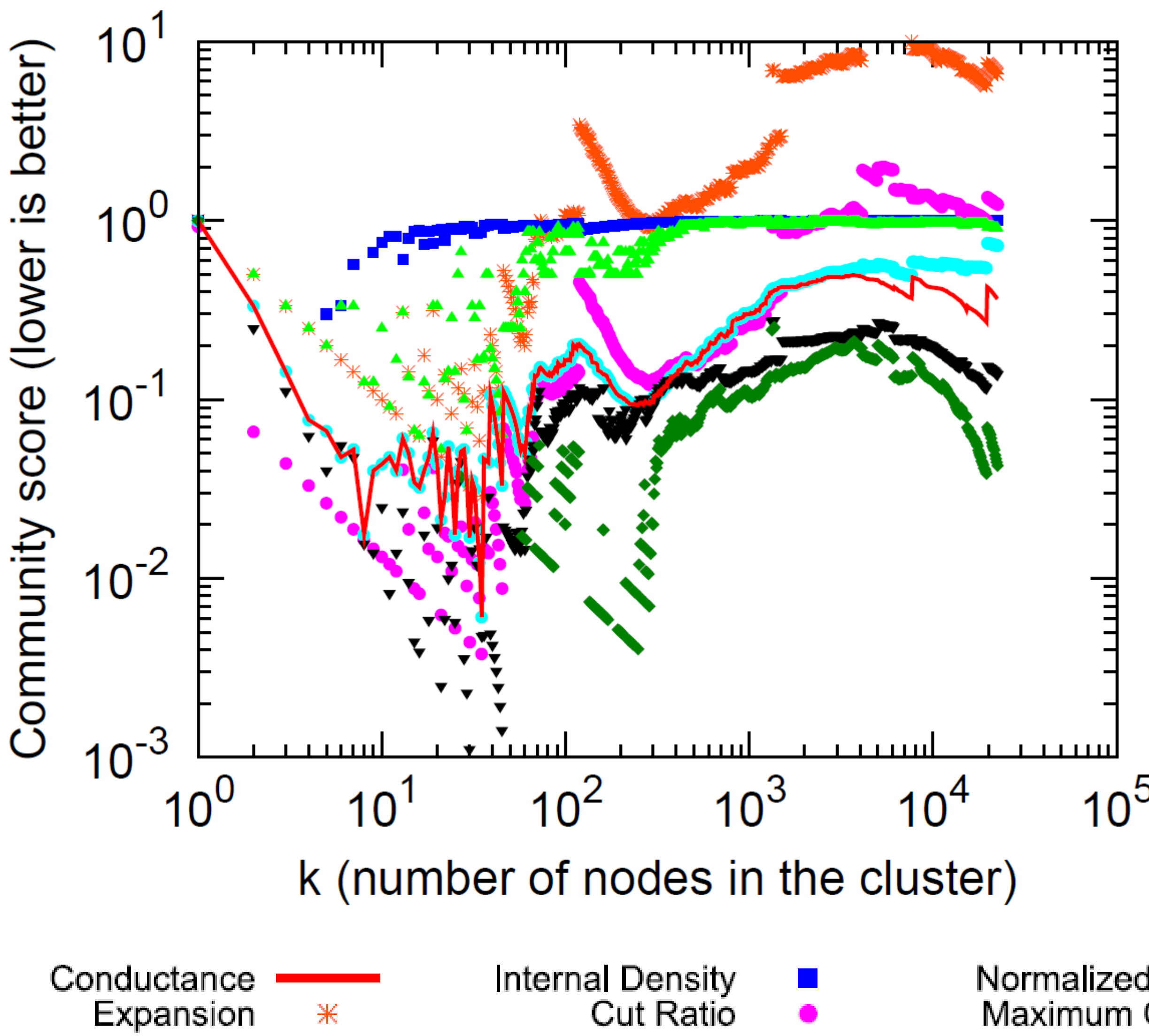






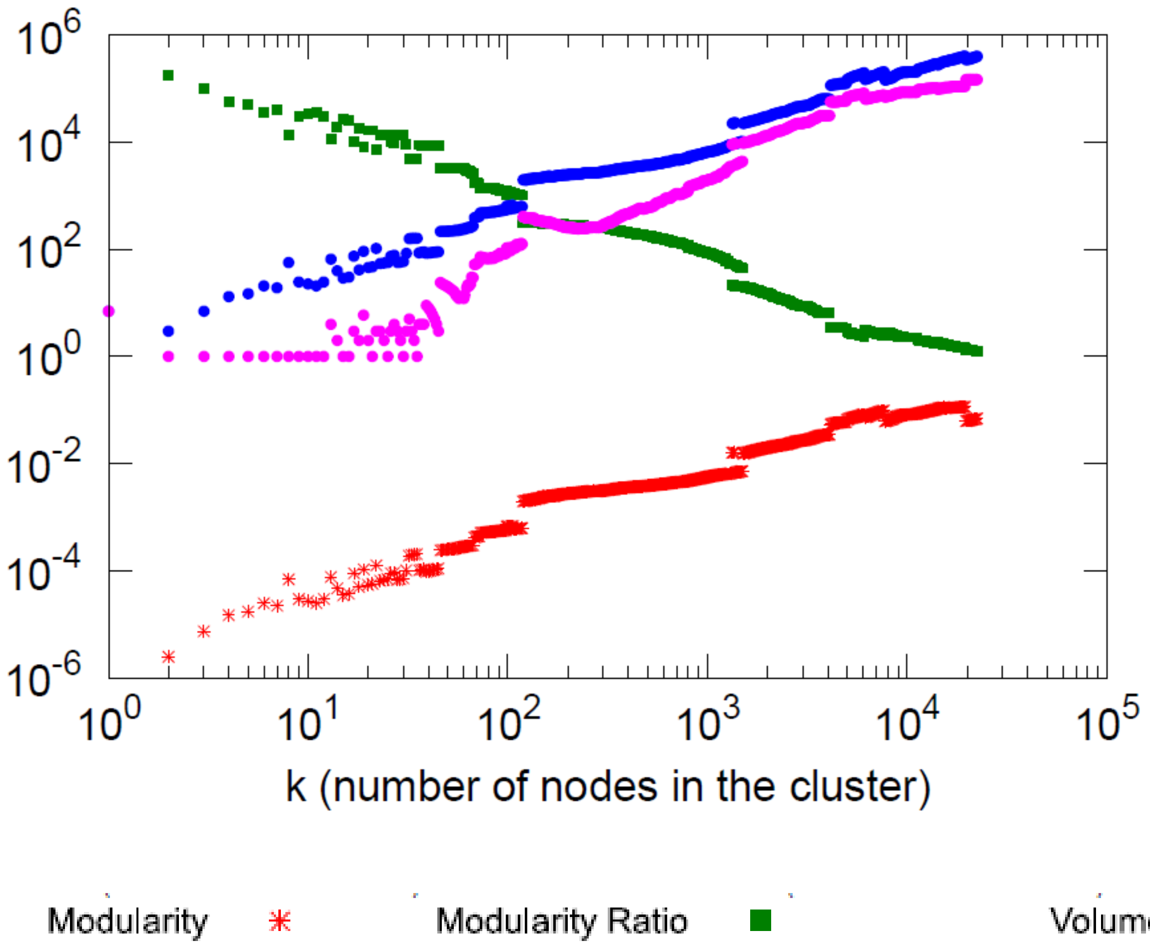
- Metis (red) gives sets with better conductance
- Spectral (blue) gives tighter and more well-rounded sets

Multi-criterion Objectives



- **All qualitatively similar**
- **Observations:**
  - Conductance, Expansion, Norm-cut, Cut-ratio are similar
  - Flake-ODF prefers larger clusters
  - Density is bad
  - Cut-ratio has high variance

Single-criterion Objectives



Observations:

- All measures are monotonic
- Modularity
  - prefers large clusters
  - Ignores small clusters